1999년도 대한전기학회 정기총회 및 추계학술대회 논문집 1999. 11. 20

이산 제어 시스템의 안정도를 위한 시간 지연의 분석

김병호, 엄광식, 서동수, 서일홍
한양대학교 전자공학과

Analysis of Time Delay for Stability of Discrete Control System

Byung Ho Kim, Kwang Sik Eom, Dong Soo Suh, Il Hong Suh
Department of Electronic Engineering, Hanyang University

Abstract - 본 논문에서는 이산 제어 시스템의 작업 수행 시, 특정작업에 있어서의 불규칙한 시간 지연 때문에 시스템이 불안정해지는 것을 막기 위해 시간 지연을 갖는 시스템의 안정도를 분석한다. Soft real time OS인 Windows NT 운영체제를 갖는 PC-based 이산 제어 시스템에서는 하드웨어적으로 인터럽트를 사용하여 시간 제한성이 있는 작업을 수행한다. 그러나 인터럽트와 함께 수반되는 DPC(Deferred Procedure Call)의 불규칙한 수행 시간 때문에 다른 작업이 수행되어야 할 표본시간이 길어지게 된다. 이러한 현상으로 다른 작업의 시간 지연이 발생하게 되며, 시간 지연은 시스템을 불안정하게 하는 요인이 된다. 안정성 분석 면에서 보면, 시간 지연을 고려하지 않은 시스템의 극점은 안정한 위치에 존재하게 되는데 반해, 시간 지연을 고려한 시스템의 극점은 불안정한 위치에 존재하게 된다.
따라서 본 논문에서는 시간 지연이 존재하는 제어 시스템의 안정성을 보장하기 위해서 시스템의 안정성을 분석한다.

1. 서 론

일반적으로 아날로그 신호로 동작을 하는 시스템의 안정성 유지와 성능향상을 위해서 디지털 컴퓨터 제어기는 적절한 주변장치를 이용하는데, 이러한 장치의 활용으로 인해서 궤한 루프에서 필연적으로 적지 않은 시간적 지연현상이 발생하게 되지만, 일반적으로 시스템에 고장이 발생하지 않고 정상적인 동작을 수행할 때에는 시간 지연을 고려한 제어기를 설계한 경우이므로 시스템의 안정성 및 성능에 미세한 영향을 미치게 된다. 그러나 이와는 다르게 내적(전체작업시간, 주기, 제어 알고리즘, 자원 제한성 등) 또는 외적(시스템이 운영되는 환경에서 발생되는 경우)인 요인으로 인해서 발생하는 제어기 고장은 상당한 시간지연과 함께 시스템의 안정성에 심각한 영향을 미치게 된다. 따라서, 시스템의 안정성 유지를 위해서 궤한 루프 상의 제어기는 다양한 고장 검출 및 고장회복알고리즘을 사용함으로써 이러한 고장을 극복하여야 한다. 하지만, 고장회복을 통한 제어기의 올바른 동작을 위해서 궤한 루프에서는 적지 않은 제어기의 계산 지연 시간이 발생하게 되고, 이것이 예상외로 길어질 경우에 시간 제한성(time-critical)을 갖는 특정작업에 있어서, 제어 시스템의 인명 및 재산상의 돌이킬 수 없는 치명적인 결과를 낳을 수 있게 된다. 이와 같이 이러한 이산 제어시스템은 시스템 동작이 제어기의 논리적인 출력 값에만 의존하지 않고 시간적인 제한성을 만족시켜야만 시스템이 안정성을 유지하면서 올바르게 동작을 할 수 있게 된다.[6] 따라서 본 논문에서는 비젼 시스템과 PC운영체제로 널리 사용되고 있는 Windows NT운영체제를 기반으로 하는 이산 제어시스템이 연동되어 로봇 시스템을 구동 시키는 작업을 하는 경우, 이산 제어시스템은 비젼 시스템으로부터의 신호를 제어하여 로봇시스템에 전달해 주는데 이산 제어 시스템 내부의 시간 지연으로 인해서 비젼 시스템으로부터의 신호를 로봇시스템에 미리 정해진 표본시간보다 지연된 시간에 전달해 줄 경우, 시스템에 미칠 수 있는 영향을 시스템의 안정성 측면에서 분석함으로써, 시스템 전체의 안정성을 보장한다.

2. Windows NT운영체제를 기반으로 하는 이산 제어시스템

2.1 Windows NT를 운영체제로 하는 이산 제어 시스템

우선 Windows NT를 운영체제로 하는 이산 제어시스템의 특징을 살펴보면 Windows NT는 멀티 스레드와 스케줄링 방식에 있어서 다중 스레드를 생성하도록 지원하며, 스케줄링 방식이 선점형이다. 그리고 실시간 운영체제를 기반으로 하는 이산 제어 시스템에서는 모든 태스크가 동일한 우선 순위를 가져서는 안되고, 우선 순위를 필요에 따라 할당해 주어야 한다. Windows NT는 스레드 우선 순위 부여에 관한 실시간 운영체제를 기반으로 하는 이산 제어시스템의 조건을 만족한다. 그러나 일반 실시간 운영체제를 기반으로 하는 이산 제어시스템 체제가 256개의 우선순위를 제공하는 반면 Windows NT 상에서 프로그램이 하드웨어에 접근하기 위한 유일한 방법은 커널 디바이스 드라이버를 통해서 이다.[5] 외부의 인터럽트에 적절히 반응하기 위해서 개발자는 커널 디바이스 드라이버를 작성해 주어야 한다. 디바이스 드라이버가 하드웨어로부터 생성된 인터럽트를 핸들링하기 위해서, windows NT는 ISR(Interrupt Service Routine)에 의해 짧은 시간 동안 서비스하고 DPC(Deferred Procedure Call)에 의해 나머지 비교적 긴 시간을 요하는 일을 처리한다. 그러나 DPC자체로는 비선점형 스케줄링 방식이라는 단점이 있다. 즉, DPC는 그것을 발생시키는 인터럽트의 우선 순위와 상관없이 모두 동일한 순위로 다루어지고 단지 ISR만이 선점형 특성을 지니기 때문에 응용 프로그램에 따라서는 다른 디바이스 드라이버에 의한 영향을 받을 수 있으며, 특히 DPC처리 시간이 긴 하드 디스크나 네트워크 드라이버에 의한 수 밀리 초의 지연을 야기 시킬 수도 있다.

3. Windows NT운영체제를 기반으로 하는 이산 제어시스템의 구성

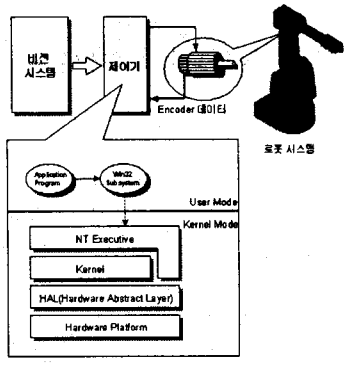


그림1 Windows NT운영체제를 기반으로 하는 이산 제어시스템

본 논문에서 제안한 Windows NT운영체제를 기반으로 하는 이산 제어시스템의 구성을 그림 1에 나타내었다. 우선, 비젼 시스템으로부터 현재 상황에 해당하는 지령을 제어 시스템으로 전달한다. 이렇게 전달된 신호는 제어기내의 Kernel mode의 HAL부분을 거쳐 User mode의 Application의 연산수행을 마친 후, 다시 HAL부분을 통해서 로봇 시스템으로 입력되고 이 값은 다시 궤환 루프를 통해 비젼 시스템으로부터 계속해서 전달되는 지령치를 보상하는 작업을 수행 한다. 여기서 제어 시스템은 windows NT를 운영체제로 하는 PC-Based 제어 시스템이다. 또한 디바이스 드라이버가 하드웨어로부터 생성된 인터럽트를 핸들링하기 위해서, windows NT는 ISR(Interrupt Service Routine)에 의해 짧은 시간 동안 서비스하고 DPC(Deferred Procedure Call)에 의해 나머지 비교적 긴 시간을 요하는 일을 처리하게 된다. 그러나 앞에서 설명했듯이 ISR은 선점형 특성을 지니기 때문에 응용 프로그램에 따라서는 다른 디바이스 드라이버에 의한 영향을 받을 수 있으며, DPC처리 시간이 긴 하드 디스크나 네트워크 드라이버에 의한 수 밀리 초의 지연을 야기 시킬 수도 있다. 따라서 그림 1에 나타난 Windows NT운영체제를 기반으로 하는 이산 제어 시스템에서 DPC처리 시간이 길어 시스템의 표본시간이 정해진 길이보다 길어지는 현상이 일어날 수 있다. 또한 DPC처리 시간은 항상 일정한 것이 아니라 불규칙하게 나타나므로 Windows NT운영체제를 기반으로 하는 이산 제어시스템 전체에 불안정한 영향을 줄 수도 있다. 그러나 일반적으로 시스템에 고장이 발생하지 않고 정상적인 동작을 수행할 때에 이는 상대적으로 작으므로 시스템의 안정성 및 성능에 미세한 영향을 미치게 될 수도 있다. 따라서 본 논문에서는 표본시간이 일정치 않아 발생하는 시간지연으로 인해서 windows NT를 운영체제로 하는 PC-Based 이산 제어시스템의 안정성을 분석한다.

4. 모의 실험

그림 1에 나타난 Windows NT운영체제를 기반으로 하는 이산 제어시스템과 동일한 시간지연현상이 일어나도록 등가적인 시스템을 설계하여 모의실험을 한다.

4.1 Windows NT운영체제를 기반으로 하는 이산 제어 시스템의 등가 시스템(표본시간이 일정치 않은 경우)

먼저, 표본시간이 일정한 continuous system에서의 상태 방정식으로부터 표본시간이 일정치 않은 시스템의 상태 방정식을 유도해야 한다. 표본 시간이 일정치 않은 경우 즉, Windows NT운영체제를 기반으로 하는 이산 제어시스템에서 ISR이 발생하여 DPC처리 시간이 길어질 경우, 표본시간이 일정치 않게 되고 이로 인한 시간지연현상이 발생하게 된다. Windows NT운영체제를 기반으로 하는 이산 제어시스템(일정치 않은 표본시간)의 등가 시스템을 구성하기 위해서는 먼저, 일정치 않은 표본시간을 시간지연으로 보고, 이 시간지연을 재분석해야 할 필요가 있다. 본 논문에서는 식 4와 같은 일정치 않은 표본시간의 재분석을 통하여 모의 실험을 하였는데, 우선, r(t)는 비젼 시스템으로부터 전달되는 지령치를 의미하며, x(t)은 상태변수, T_s(0.001[s])는 표본시간, H(z)는 PI제어기, E는 e(t)의 Z-Transform이며, U도 로봇 시스템에 전달되는 입력 u(t)의 Z-Transform이다. A, B, C는 상태 계수행렬을 의미하며,A=-6.5,B=1,C=6.5이다.($K_P=1000, K_I=100$

$$r(t)=\begin{cases} t & (0[s]\le t<6[s]) \\ 6 & (6[s]\le t\le 10[s]) \end{cases} \quad (1)$$

$$H(z)=K_p+\frac{1}{2}\frac{K_i}{Z-1} \quad (2)$$

) 그리고 Delay ΔT_i들의 평균값을 ΔT_{av}으로 정하고, 표본시간 T_s를 더하여 KT^*를 정의한다. 또한 ΔT_{av}은 메 표본시간마다 일정한 값이므로 KT^*도 매 표본시간마다 일정한 값을 갖는. 그리고 일정치 않은 매 표본시간의 편차 즉, $\Delta T_i-\Delta T_{av}$을 δT_i으로 정의한다. 이러한 분석을 통하여 시스템의 안정성을 분석할 때, 편차의 값들이 평균값을 중심으로 해서 음수와 양수의 값으로 나타나기 때문에 시스템 안정성 분석을 보다 용이하게 해준다. 식 4와 같은 재분석을 통해서, 모의 실험을 위한 상태방정식을 유도한다. 먼저, 표본시간이 일정한 continuous system에서의 상태 방정식 식 (1)로부터 표본시간이 일정치 않은 시스템의 상태 방정식을 유도해야 한다. 다음에 나타나는 수식들은 유도하는 과정을 보인다. 아래와 같은 유도과정을 통해서 그림 3과 같은 Windows NT운영체제를 기반으로 하는 이산 제어시스템(일정치 않은 표본시간)의 등가 시스템을 구성할 수

- Continuous Time

$$x(t) = e^{A(t-t_0)}x(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau \qquad (3)$$

- Discrete Time(일정치 않은 sample time일 경우)

$$X[(T_s + \Delta T_1) + (T_s + \Delta T_2) + \ldots + (T_s + \Delta T_k) + (T_s + \Delta T_{k+1})]$$
$$= X[(k+1)T_s + \sum_{i=1}^{k+1} \Delta T_i]$$
$$= X[(k+1)(T_s + \Delta T_{av}) + \delta\tilde{T}_{k+1}]$$
$$= \boxed{X[(k+1)T^* + \delta\tilde{T}_{k+1}]}$$
$$= e^{A(T^*+\delta\tilde{T}_{k-1}-\delta\tilde{T}_k)}X[kT^* + \delta\tilde{T}_k] + \int_{kT^*+\delta\tilde{T}_k}^{(k+1)T^*+\delta\tilde{T}_{k+1}} e^{A((k+1)T^*+\delta\tilde{T}_{k+1}-\tau)}Bu(\tau)d\tau$$
$$= e^{A(T^*+\delta\tilde{T}_{k+1})}X[kT^* + \delta\tilde{T}_k] + e^{A((k+1)T^*+\delta\tilde{T}_{k+1})}\int_{kT^*+\delta\tilde{T}_k}^{(k+1)T^*+\delta\tilde{T}_{k+1}} e^{-A\tau}Bu(\tau)d\tau$$
$$\boxed{= e^{A(T^*+\delta\tilde{T}_{k+1})}X[kT^* + \delta\tilde{T}_k] + e^{A((k+1)T^*+\delta\tilde{T}_{k+1})}\int_{kT^*+\delta\tilde{T}_k}^{(k+1)T^*+\delta\tilde{T}_{k+1}} e^{-A\tau}Bd\,u[kT^* + \delta\tilde{T}_k]}$$

$$\boxed{Y[kT^* + \delta\tilde{T}_k] = Cx[kT^* + \delta\tilde{T}_k]} \qquad (4)$$

있다. 매 표본시간마다 불규칙한 δT_i값을 변화 시키면서 출력을 분석하였다. 모의 실험 결과는 그림 3과 같다.

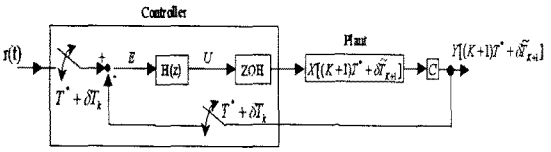
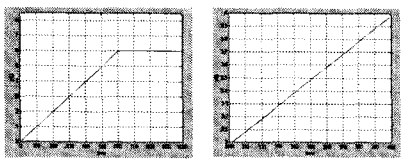
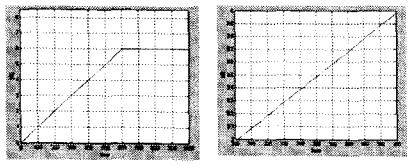


그림 2. Windows NT운영체제를 기반으로 하는 이산 제어시스템의 등가 시스템
(일정치 않은 표본시간)

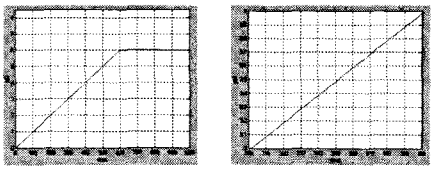
그림 3에서도 알 수 있듯이 표본시간의 편차 δT_i를 변화 시키면서 얻은 출력 결과가 그래프 전체의 모습으로는 지령 기준치와 거의 같은 모습으로 나타났고, 자세한 분석을 위해서 300[s]~400[s]동안의 출력결과를 분석하였다. 분석 결과는 그림 3에서와 같이 표본시간의 편차 δT_i의 크기를 표본 시간내에서 변화를 시키더라도 시스템 전체를 불안정하게 만들지는 않는다는 것을 알 수 있다.



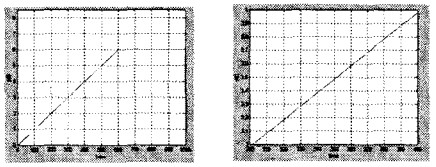
(a) $\Delta T_{av} = 0, \delta T_i = 0$



(b) $\Delta T_{av} = \frac{T_{av}}{4}, \delta T_i = 0$



(c) $\Delta T_{av} = \frac{T_s}{4}, -\frac{1}{2}T_{av} \le \delta T_i \le \frac{1}{2}\delta T_{av}$



(d) $\Delta T_{av} = \frac{T_s}{4}, -T_{av} \le \delta T_i \le T_{av}$

그림 3. 모의실험 결과

5. 결 론

지금까지 보았듯이 본 논문에서는 비젼 시스템과 PC운영체제로 널리 사용되고 있는 Windows NT운영체제를 기반으로 하는 이산 제어시스템이 연동되어 로봇 시스템을 구동 시키는 작업을 하는 경우, ISR이 발생되면 그로 인한 DPC처리 시간이 길어져 실제로 시스템 전체의 안정성에 어느 정도 영향을 주는지에 관한 분석을 해보았고 그 결과 시스템 전체가 안정함을 보였다. 따라서 본 논문에서 연구한 Windows NT운영체제를 기반으로 하는 이산 제어시스템에서는 ISR으로 인해 DPC가 길어질 경우에 시스템의 안정성이 보장됨을 보였다.

[참 고 문 헌]

[1] G.C. Buttazzo, and M. D. Natale, "HARTIK:A Hard Real-Time Kernel for Programming Robot Task with Explicit Time Constraints and Guaranteed Execution", IEEE International Conference on Robotics and Automation, pp.404-409,1993

[2] Mohammad Jamshdi,Manu Malek-Zavarei, Time-Delay System Analysis, Optimization and Application pp23-42, 1995

[3] S. McConnel, D. Siewiorek, and M. Tsao, The measurement and analysis of transient errors in digital computer systems, Digest of Papers, FTCS-9, pp. 67-70, June, 1979

[4] K.Shin and H. Kim, Derivation and application of hard deadlines for real-time control systems, vol. 22,no. 6, pp.1403-1413, November, 1992

[5] S. W. Son, And K. D. Lee,An Implementation and Performance Analysis of Real_Time Robot Control Software,Trans. KIEE. Pp1264-1272, vol. 46, NO. 8, Aug, 1997

[6] 김학배, 이대현,다중샘플링 다중작업을 수행하는 실시간 제어시스템의 시계수제한성 유도,Journal of Control, Automation and Systems Engineering, Vol 5, NO. 2, February, 1999